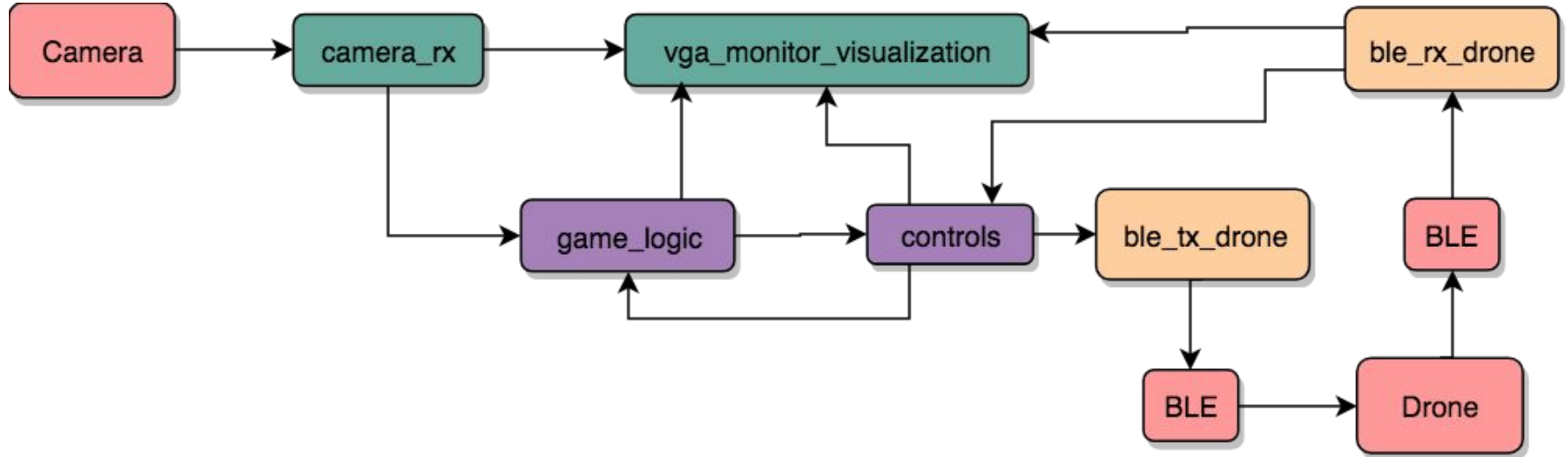# AirPong

Bret Heaslet, David Mueller, Sabina Chen
6.111 Fall 2018

AirPong Overview

"Introducing *AirPong*... where players are the paddles and a drone is the puck."

- Camera to track real-time locations of drone and players
- BLE communication between FPGA and drone
- Self-defined controls
- Augmented VGA monitor visualization

Block Diagram

The Quadcopter

- Parrot Mambo Mini-drone
    - 8 inch x 8 inch
    - Open-source
    - Bluetooth communication

Modules - BLE TX/RX

- Communications modules intended to interface with a UART BLE module.
  - ble_rx module will receive and interpret sensor data from the drone, via the BLE module. The interpreted data is packetized and stored in a 32-bit register and forwarded on to the controls, and computer vision modules.
  - ble_tx module will interpret commands from the controls module and transmit interpreted, packetized commands to the BLE module. The BLE module then forwards the command to the drone.
- Hardware
  - We will be utilizing the Bluetooth 4.0 HM-10 module.

Modules - Camera RX

- Camera module that processes pixel data from the NTSC camera and uses chroma keying to identify locations of drone and paddles
  - Input: pixels
  - Ouput: x and y coordinates of the center of drones and paddles
- Hardware
  - NTSC Camera
  - Wired directly to FPGA - will be hung from the ceiling to get a top-down view of game
  - Field of view: 700 x 525

Modules - VGA Monitor Visualization

- 2D Visualization module on the VGA monitor that will display game state information in an augmented format
  - <u>Inputs</u>: x and y coordinates of drone/paddles (via camera_rx), game state info (via game_logic), flight telemetry (via ble_rx/controls)
  - <u>Outputs</u>: Augmented 2D game visualization of the pong game overlayed on a live video feed from camera on the vga monitor
- Goal: Display game logic and game state in action
- Useful for debugging other modules!

Modules - Game Logic

- FSM module used to keep track of game state
    - Almost completely re-used from pong lab
    - Original implementation was written for modularity
- Continuously sends commanded velocity to the controls module
- Keeps track of player scores and win/loss conditions

Modules - Controls

- Receives commanded velocity from the Game Logic module
- Provides speed commands for all 4 drone motors.
- Speed commands obtained from feedback controller using:
  - Position, altitude, climb rate, roll, pitch, yaw
- Position information is obtained from overhead camera
- Acceleration and other state information obtained from onboard sensors
- Also controls takeoff and landing sequence
- Controller to be designed as part of joint project with 16.30's final project.

Timeline:

	11/4							11/11							11/18							11/25							12/2							12/9			
	S	M	T	W	Th	F	Sa	S	M	T	W	Th	F	Sa	S	M	T	W	Th	F	Sa	S	M	T	W	Th	F	Sa	S	M	T	W	Th	F	Sa	S	M	T	W	Th	F	Sa
ble_rx_drone																																				
Implement serial rx		■	■	■	■																															
Build interpreter from sensor specs		■	■	■	■																															
Test/Debug ble_rx_drone with BLE module						■	■	■	■	■	■																									
Receive data from drone									■	■	■	■																								
Integrate with other modules												■	■	■	■	■	■																			
ble_tx_drone																																				
Build interpreter from drone commands		■	■	■	■																															
Implement serial tx		■	■	■	■																															
Test/Debug ble_tx_drone with BLE module						■	■	■	■	■	■																									
Transmit commands to drone									■	■	■	■																								
Integrate with other modules												■	■	■	■	■	■																			
camera_rx																																				
Get color projection from camera		■	■	■																																
Color recognition via chroma keying					■	■	■																													
Object/Color blob recognition						■	■	■																												
vga_monitor_visualization																																				
Overlay pong/paddles over live video							■	■	■																											
Visualize game states as they are developed									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■					
game_logic																																				
Integration & Testing									■	■	■	■	■	■	■	■	■	■	■	■																
controls																																				
Control Law Synthesis		■	■	■	■	■	■	■	■	■	■	■																								
Control Law Testing							■	■	■	■	■	■					■	■	■	■	■															
Control Law Integration																■	■	■	■	■																
Testing & Tuning																		■	■	■	■	■	■	■	■	■	■	■	■							
Project Checkoff																																				
Video Recording Due																																				
Project Report Due																																				

Questions?